

Monitor Anything

I. Dimou

What is monitoring?

Aka:

Logging

Observability

Monitoring

Telemetry

Event Management

Not this event management...



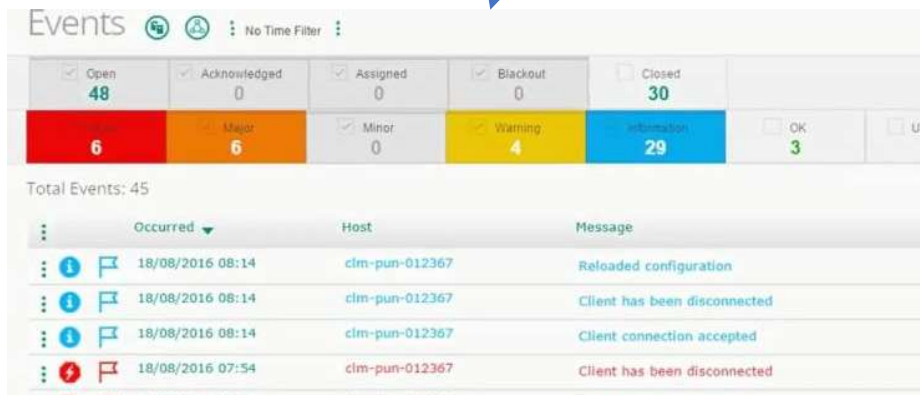
Big Data

κανείς δε μπορεί να κάνει big data χωρίς ένα γερό collection mechanism.

Why?

To catch issues before the users see them.

=> operate on the **event space** instead of the **incident/ticket space**.



The screenshot shows a dashboard titled 'Events' with a 'No Time Filter' option. It features a summary grid with the following counts:

Open	Acknowledged	Assigned	Blackout	Closed
48	0	0	0	30

Below this, there is a row of event categories:

Minor	Warning	Information	OK	Unknown
6	4	29	3	0

The total number of events is 45. A table below lists recent events:

Occurred	Host	Message
18/08/2016 08:14	clm-pun-012367	Reloaded configuration
18/08/2016 08:14	clm-pun-012367	Client has been disconnected
18/08/2016 08:14	clm-pun-012367	Client connection accepted
18/08/2016 07:54	clm-pun-012367	Client has been disconnected



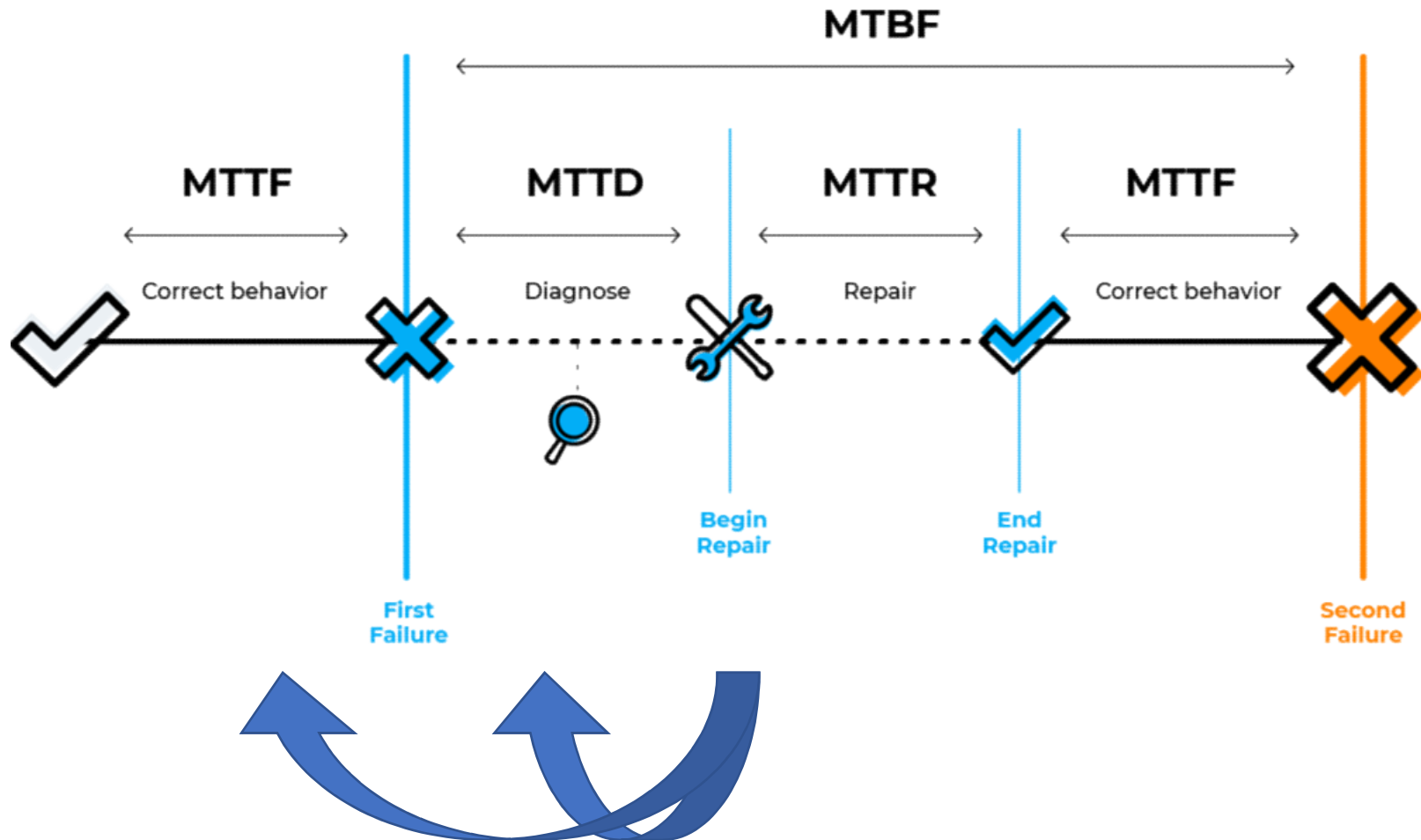
Also

to do SLA reporting

to measure application feature usage

to audit

A bit of systems engineering

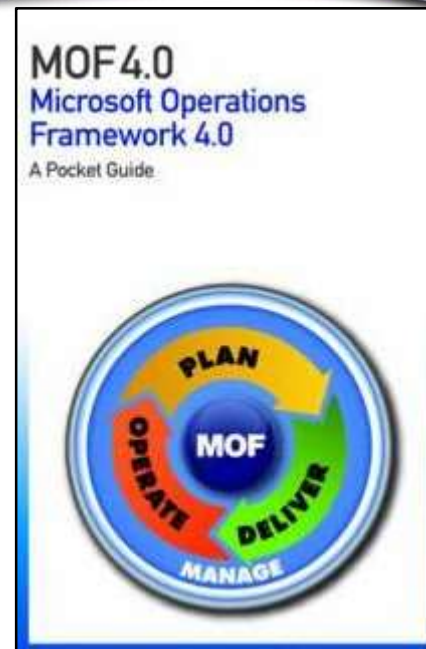


Processes = automation in the people domain

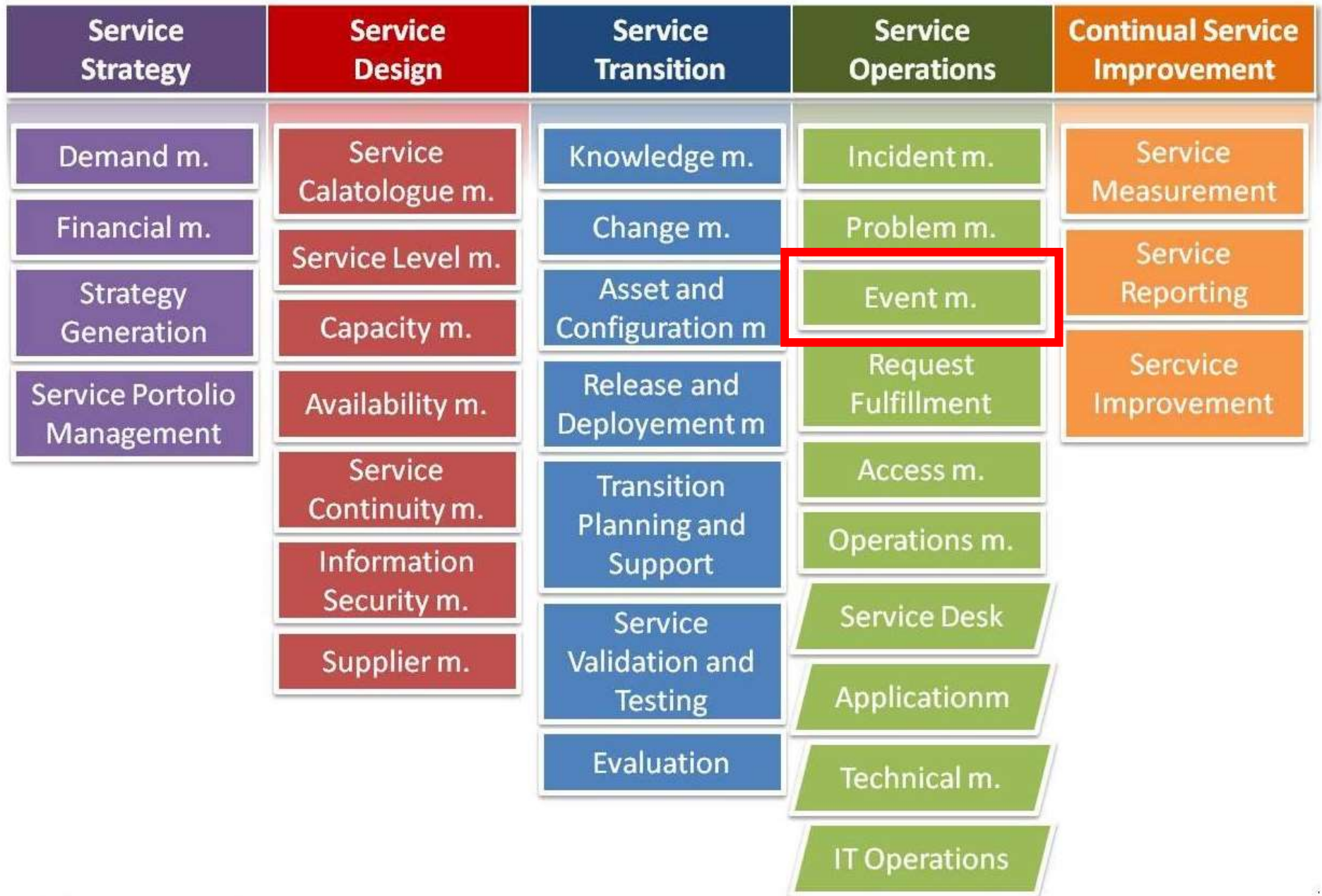
IT Process frameworks:

ITIL

Microsoft Operations Framework



ITIL Process Map



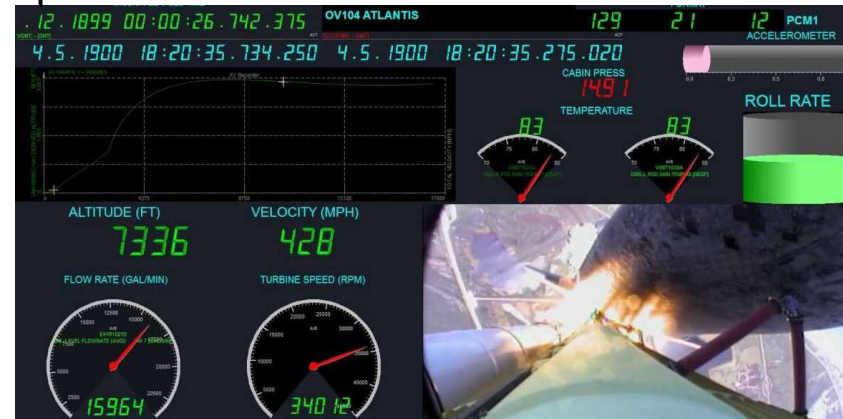
How?

With a monitoring system fit for what we need to monitor.

Hospital



Spacecraft

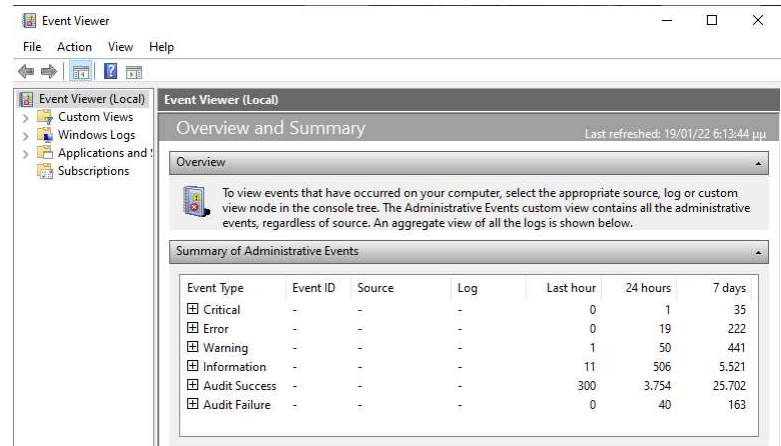


Solar system



Device ID	Alert Class	Alert Code	Occurred
10017250 90342KB40001C	Inverter: Grid Fault	240	1/4/2022, 8:59:59 AM
10017250 90342KB40001C	Inverter: Grid Fault	240	12/19/2021, 8:44:59 AM
10017250 90342KB40001C	Inverter: Grid Fault	240	12/9/2021, 8:44:57 AM
10017250 90342KB40001C	Inverter: Grid Fault	240	12/3/2021, 8:30:00 AM
10017250 90342KB40001C	Inverter: Grid Fault	240	12/2/2021, 8:29:59 AM

Computer



From raw events to smart alerts



High volume
Low quality
Streaming. Short retention times.

A few
Actionable
Enriched with context

The Event Management Pyramid

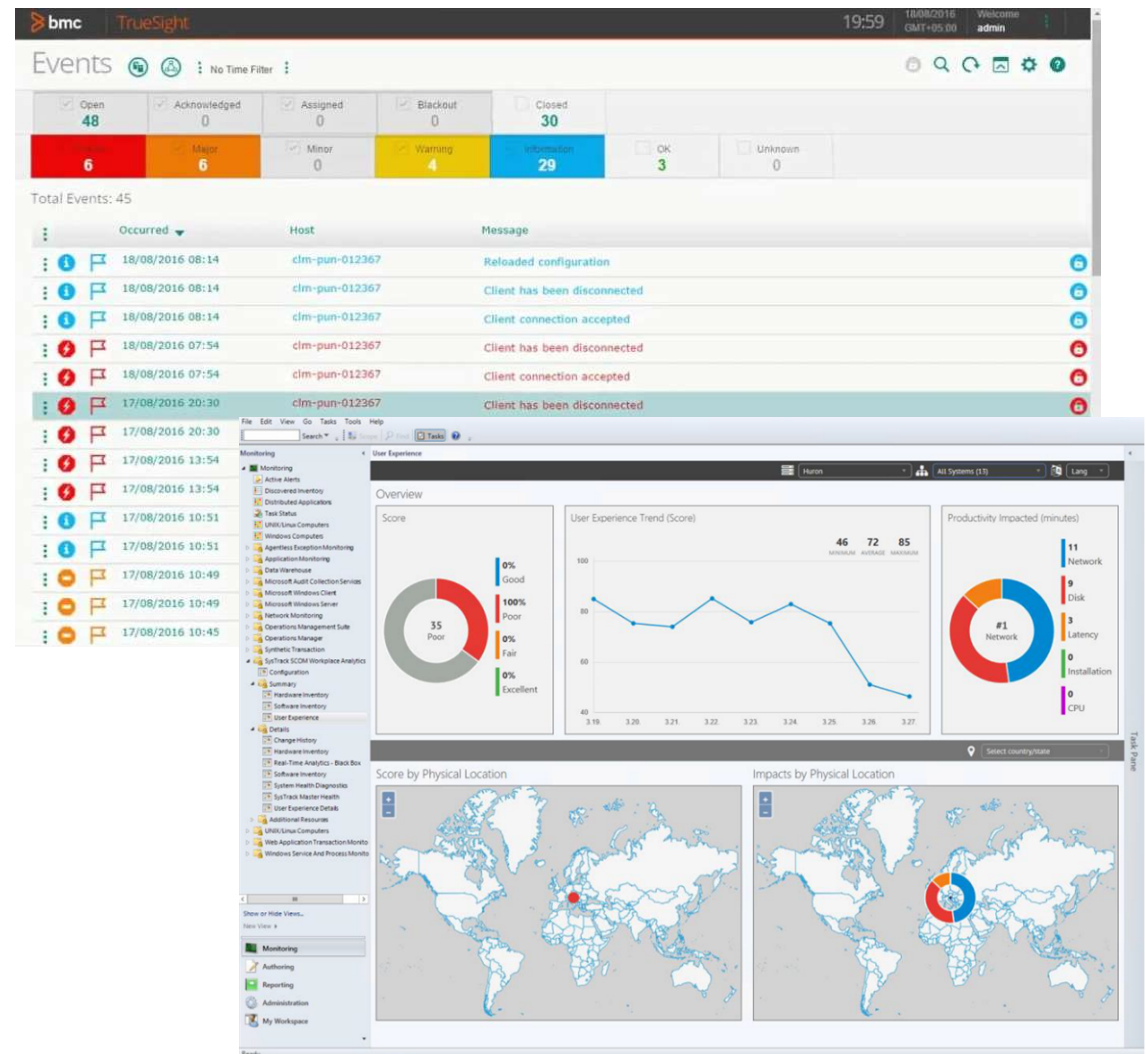


Enterprise Level Tooling & design

Complex-Expensive-Large Scale
Need Specialists to set up

ServiceNow ITSM
BMC TrueSight
IBM Netcool
Microsoft SCOM

Integrated with
CMDB's service models
they can show incident impact.

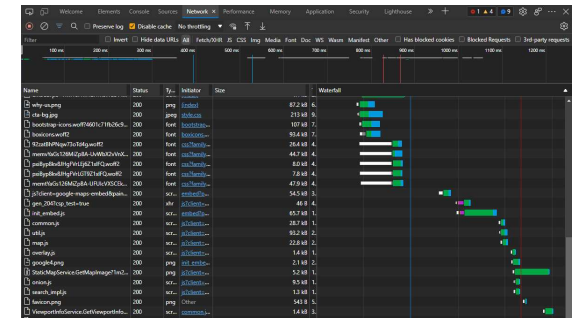


Application Performance Monitoring (APM)

How to do it wrong: Monitoring storage, CPU, memory utilization

How to do it right:

1. Break down the app into user identifiable functions.
2. Monitor user experience based on real transactions
3. Traceability from client to app to DB to infra (also cool maps)
4. Call traces (like browser dev tools)

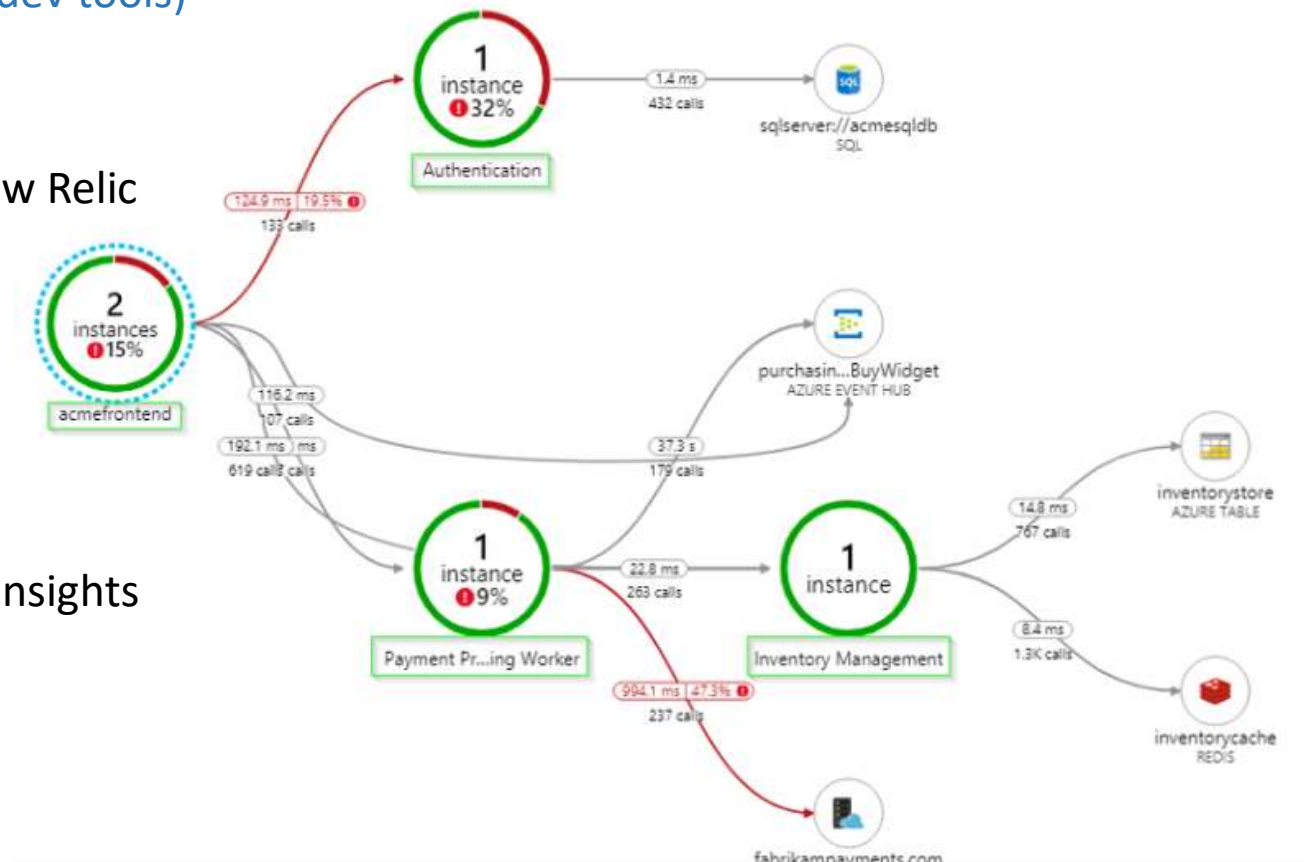


Enterprise solutions:

Dynatrace, AppDynamics, New Relic
(>5000€/year/node)

Value for money:

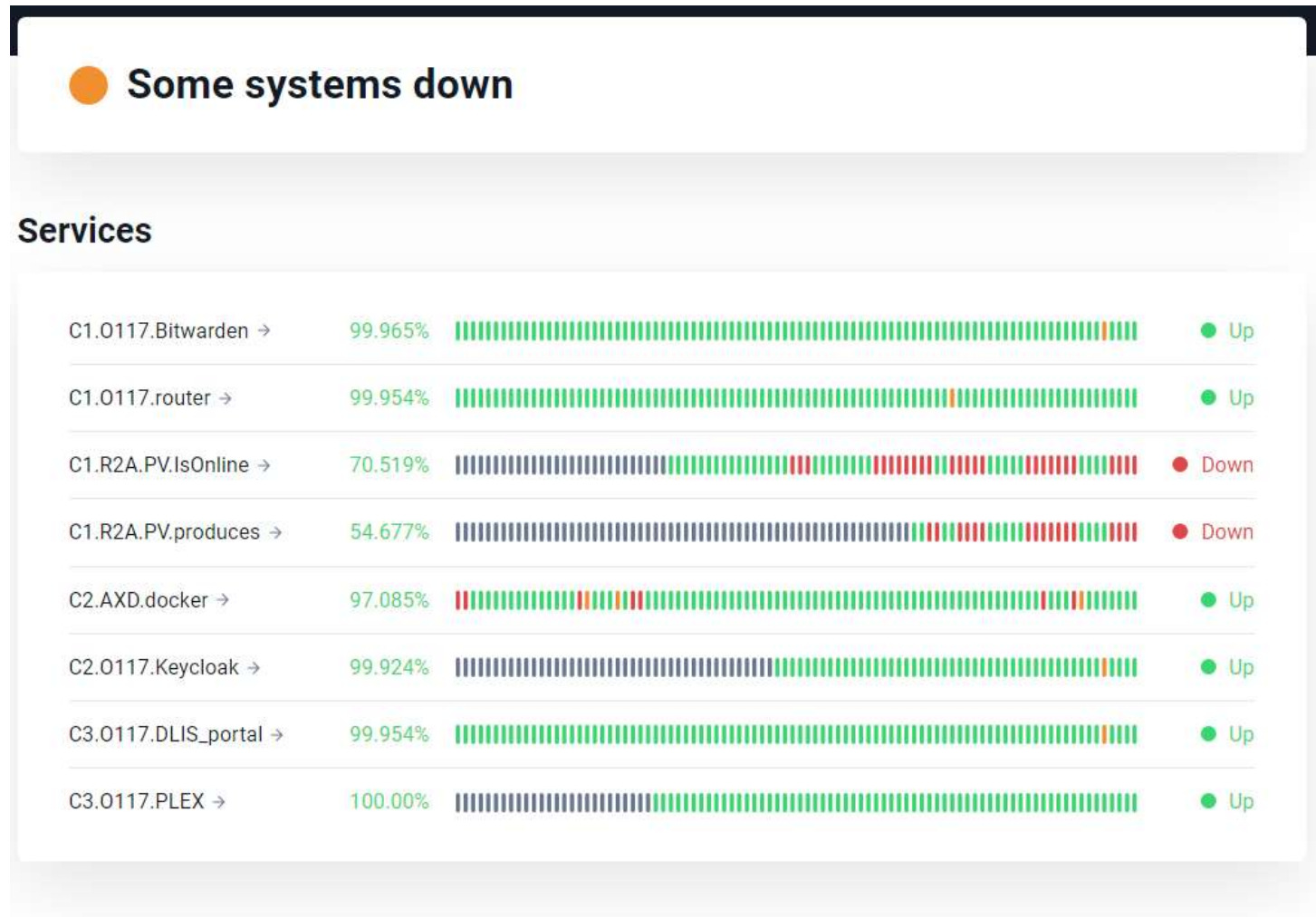
Microsoft Azure Application Insights



Minimal internet based: UptimeRobot free

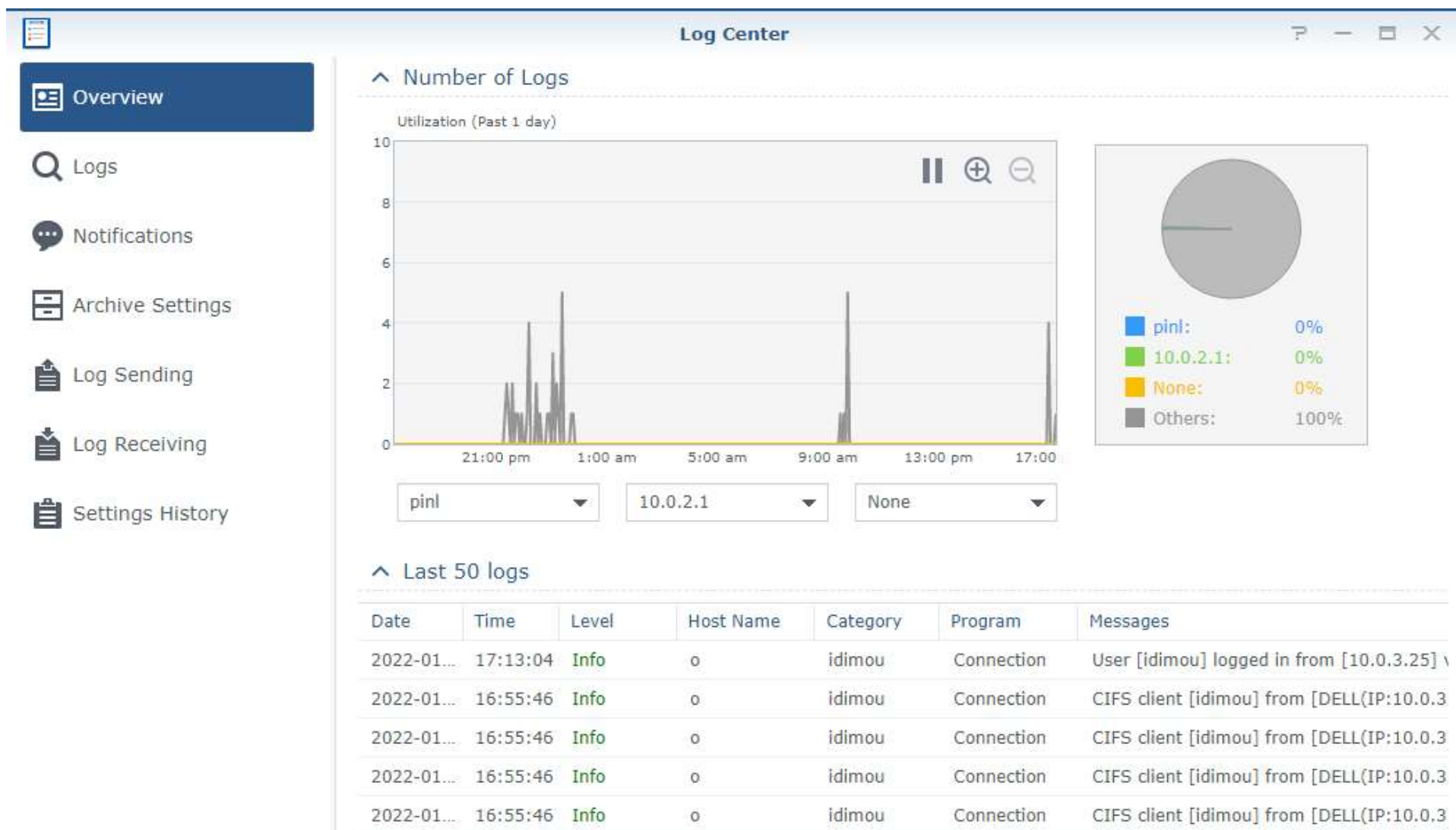
Setup in <30'

Can also alert on mobile/email



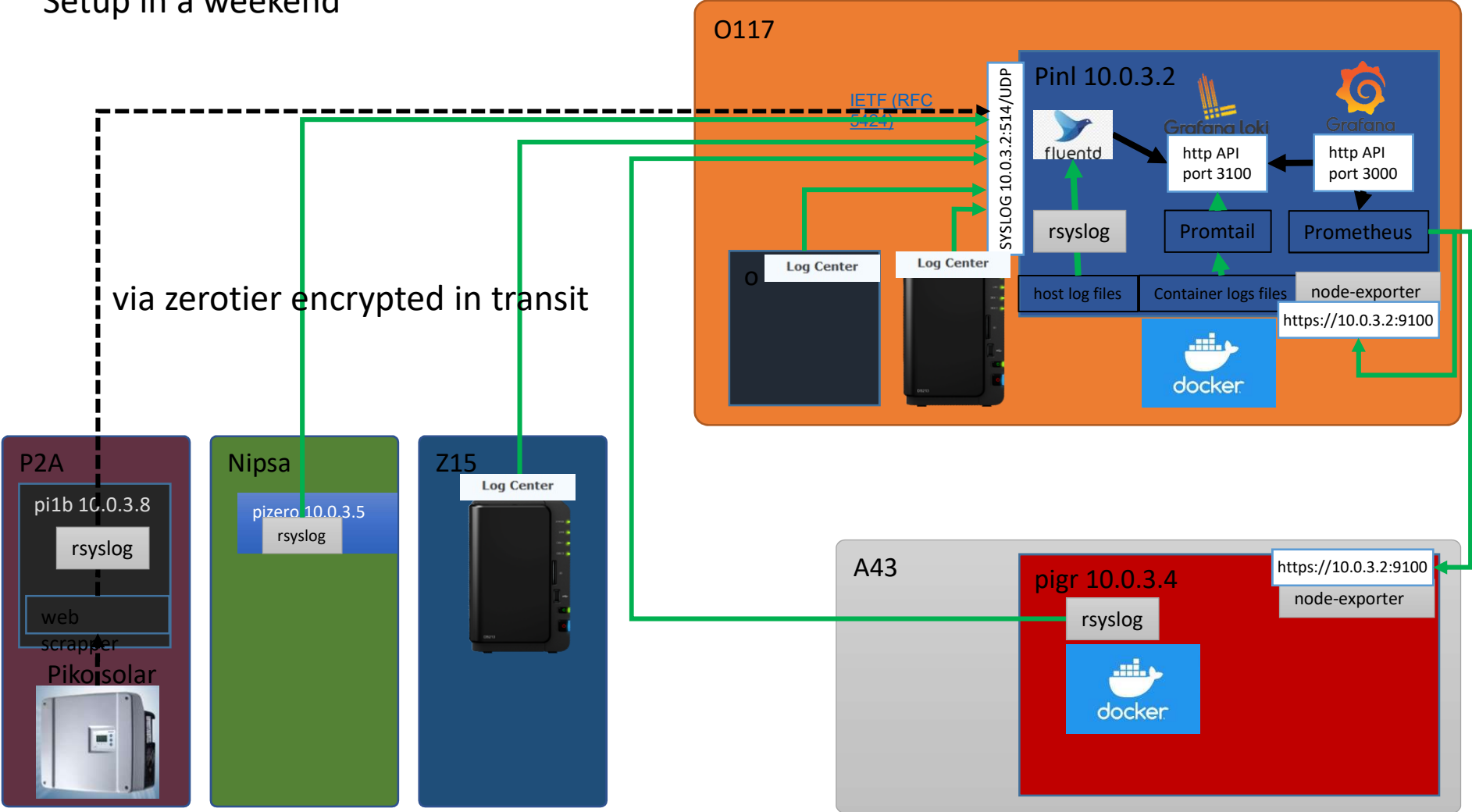
Minimal self hosted: Synology Log Center

Setup in <30'



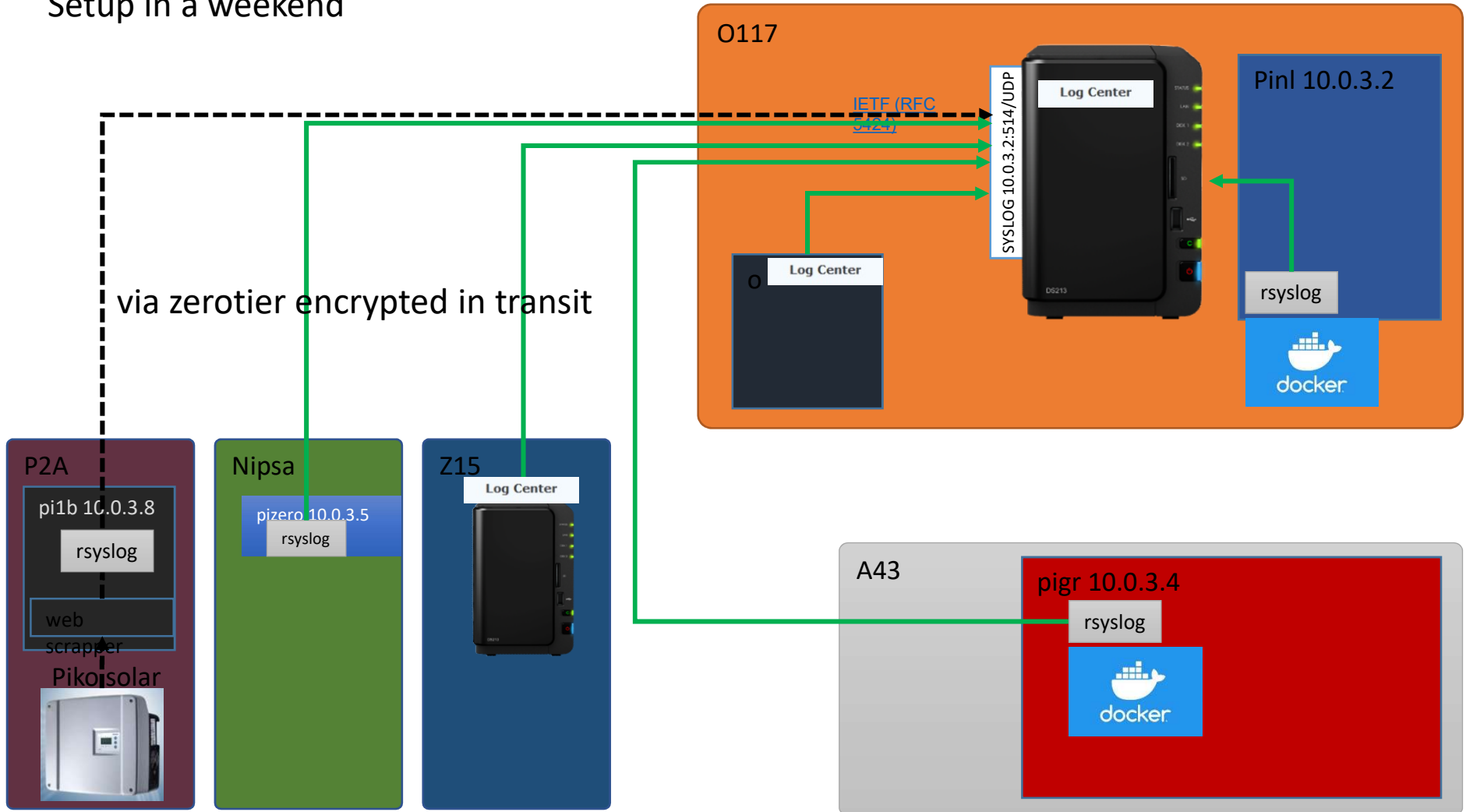
SME/Home lab level monitoring setup

Setup in a weekend



SME/Home lab level monitoring setup

Setup in a weekend



Monitoring a Linux host via node-exporter + Prometheus



Alerting

Grafana alerts to Discord channel

Starfleet's server

telemetry

1 new message since 17:40

Mark as read

[Alerting] Test notification

Someone is testing the alert notification within Grafana.

High value
100.000000

Higher Value
200.000000

Memory / CPU

Time	memory (B)	cpu
15:40	15.0	2.0
15:50	10.0	1.5
16:00	15.0	2.0
16:10	10.0	1.5
16:20	5.0	1.0
16:30	10.0	1.5

Grafana v7.5.4

2 October 2021

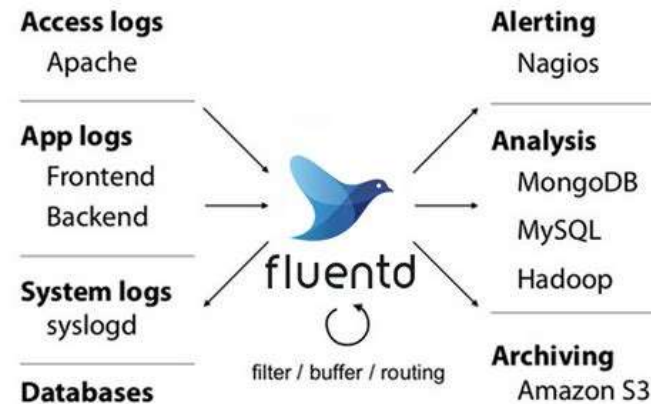
Grafana BOT 02/10/2021

[OK] Bitwarden events alert

Standardization is key

Standard collectors

Use whatever **agent** is available and standardize e.g. via **fluentd** collector.



Standard protocols (this is a real mess)

Pick one that suits you e.g. the latest [SYSLOG IETF RFC5424](#) over UDP

For apps you can do the same. The content matters not the envelope.

Standard semantics

Log severity levels should mean the same everywhere

Metric definitions e.g. availability intervals



A collector

fluentd.conf

C: > Users > ioann > Desktop > fluentd.conf

```
1 <source>
2 @type syslog
3 tag syslog.logs
4 port 24224
5 bind 0.0.0.0
6 @log_level info
7 <parse>
8   @type syslog
9   message_format auto
10  with_priority true
11 </parse>
12 emit_unmatched_lines true
13 </source>
14
15 <match **>
16 @type loki
17 url "http://telemetry_loki:3100"
18 flush_interval 5s
19 flush_at_shutdown true
20 buffer_chunk_limit 1m
21 extra_labels {"job":"syslog"}
22 </match>
```

Tag events to filter later

Parse type to get all properties

What to send

Where to send

Label outgoing events



Things to watch out for



- Correct clocks and time zones in all monitored systems. Otherwise Loki will discard events. →
- Non-blocking docker container monitoring only via log file scrapping. →
(other docker logging drivers freeze the container if they cannot log)
- Make sure severity levels and host names are mapped correctly. Makes filtering easy. →


```
/ $ date
Sun Jan  0 00:100:4174038 1900
/ $ █
```

```
network_mode: backend
restart: unless-stopped
logging:
  driver: "json-file"
  options:
    max-size: '50m'
    max-file: '3'
    tag: '{{.Name}}'
```

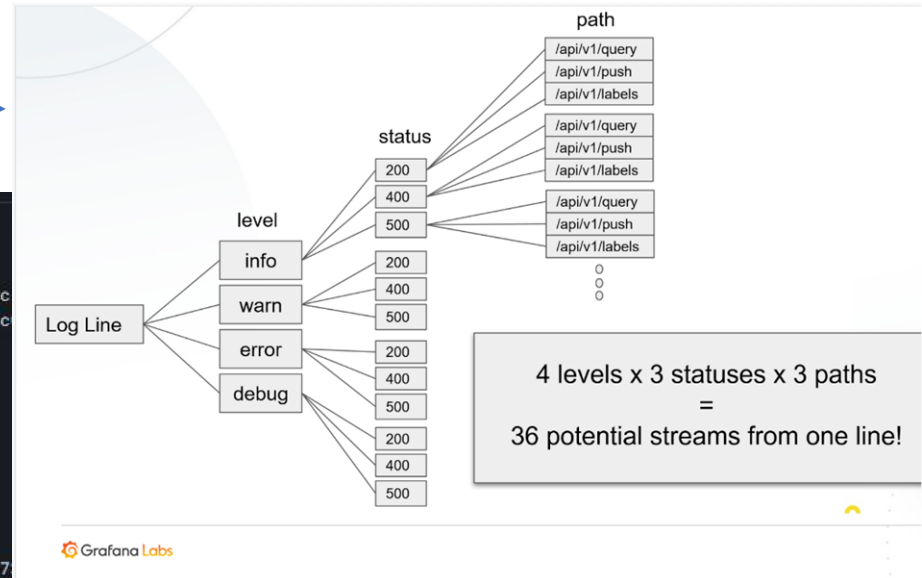
Severity
0-Emergency
1-Alert
2-Critical
3-Error
4-Warning
5-Notice
6-Informational
7-Debug



More things to watch out for

- Keep it lean and fast (no big indexes):
e.g. use Grafana's event collector 
with no indexes and very few properties.

```
Log labels
  container_name telemetry_promtail
  filename /containerslogs/08a0428d487523a95bac7523a95bac52b9aee7d3913ca2878e32fa1c
  job containerlogs
Detected fields
  caller tailer.go:99
  component tailer
  level info
  msg "position timer: exited"
  path /containerslogs/d81776c5a07bbade1f977bbade1f973bb0feba461a85bbdd55641005
  ts 2022-01-20T12:53:35.024Z
```



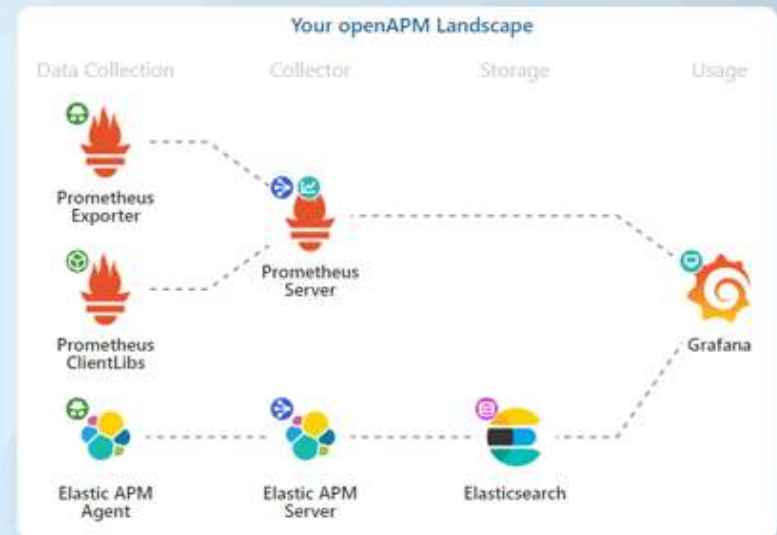
- Make it reliable/survivable
 - Minimize dependencies: Can work without DNS, complex routing, etc.
 - Use simple architecture
 - Run it on a separate node doing only this task
 - **The monitoring system needs to work when nothing else does.**
- Security
 - Sanitize logs or encrypt in transit

Component selection & design



OpenAPM

Your custom open source APM solution



Take aways

1. Think first what you want to monitor.
 - Is it worth setting up a monitoring capability? Will it save you time/money?
2. Pick the right tool(s). <https://openapm.io>
 - Inputs: Will you be able to connect all the inputs that you need?
 - Is it easy to configure?
 - Outputs: Can it feed the dashboards that you want?
3. Invest time in good rules. It will save you a lot of time later.
4. Improve incrementally. Mute noise, enrich, create good actionable alerts.

[Google - Site Reliability Engineering \(sre.google\)](https://sre.google)

[ITIL | IT Service Management | Axelos](https://www.axelos.com)

[The global skills and competency framework for a digital world — English \(sfia-online.org\)](https://sfia-online.org)

